Question 1
What do these acronyms mean?
(a) CPU?
(b) ROM?
(c) EPROM?
(d) RWM?
(e) RAM?
(f) I/O?

Question 2
What role does the CPU play in a computer system?

Question 3
Why is ROM an essential part of any computer and when is it most needed?

Question 4
Explain the difference between writing to memory and writing to I/O devices.

Question 5
How many bits are there in:
(a) a byte?
(b) A word?

Question 6
Convert 57 (decimal) into an 8-bit binary number. What is the value of:
(a) its most significant bit (MSB)?
(b) Its least significant bit (LSB)?
(c) Its low nibble (a hex digit)?

Question 7
Using binary arithmetic, evaluate:
(a) 01110 + 00101
(b) 01110 – 00101
Question 8
Using bitwise logic, evaluate:
(a) 01110 AND 00101
(b) 01110 OR 00101

Question 9
Using bitwise arithmetic, evaluate:
(a) 6 or 7
(b) 2 AND 5

Question 10
If the address on the address bus were 0010 0000 0000 0000, what memory location (hexadecimal number) would be accessed by the microprocessor?

Question 11
What is the maximum memory that can be accessed by 16 address lines?

Question 12
The Hitachi HM6116 Static CMOS RAM chip has 11 address lines (bits) and 8 data lines.
(a) Is this a CPU, memory or I/O chip?
(a) What is its storage capacity in kB?

Question 13
A high density 3 1/2 inch floppy disk has 2 sides, 80 tracks or cylinders, 18 sectors per track and 512 bytes per sector.
(a) What is its capacity in kilobytes?
(b) What is its capacity in megabytes?
(c) MS-DOS makes 2847 sectors available for user data. The remainder is reserved for system use such as the boot sector, the file allocation tables and the root directory. How many kB are reserved?

Question 14
The original IBM PC/XT computer has a 20 bit address bus (i.e. addresses have 20 bits).
(a) How many hex digits are needed to specify an address?
(b) How much memory can an IBM PC/XT address? Give your answer in megabytes.
(c) If 640K RAM is available for user programs, how many kB are reserved for system use?
Question 1
(a) Convert $101010_2$ into a positive decimal number.
(b) Convert $8FF_{16}$ into a positive decimal number.
(c) Convert $2337_{10}$ into a positive 4-digit hexadecimal number.
(d) Convert $29_{10}$ into a positive 6-bit binary number.
(e) Convert $9A8_{16}$ into a positive binary number.
(f) Convert $100111001110_2$ into a positive hexadecimal number.

Question 2
(a) Convert $7_{10}$ into an 8-bit two’s complement binary number.
(b) Convert $-7_{10}$ into an 8-bit two’s complement binary number.

Question 3
(a) What is the ASCII code for the letter “F”?
(b) What string has the sequence of ASCII codes “36 38 30 30 30 00”?
   (All codes are in hexadecimal representation)

Question 4
Express the following in 32-bit floating point representation
(a) $5.75_{10}$
(b) $313.125_{10}$
(c) $-54.625_{10}$

Question 5
Does $0.1$ have an exact representation in a 32-bit floating point format? Justify your answer.

Question 6
Assume that the following hexadecimal numbers represents single-precision floating point numbers:
(a) C1A4C000
(b) BE180000
(c) 4332A000
Determine their decimal equivalents.
Question 1

Suppose that data register, (D0) = 0x00000000 and word location
(700000) = 0x8765
(700002) = 0x4321
What is the content of D0 after executing the following:
(a) MOVE.B 0x700000, D0
(b) MOVE.W 0x700000, D0
(c) MOVE.L 0x700000, D0

Question 2

Suppose that data register (D0) = 0x0101 0101 and the content of the memory
location (5010) = 0x87, (5011) = 0x65, (5012) = 0x43, (5013) = 0x21.
What is the content of data register, D0 after the execution of:
(a) ADD.B 0x5010, D0
(b) ADD.W 0x5010, D0
(c) ADD.L 0x5010, D0

Question 3

What is the difference between the following two instructions:
MOVE.B 0x1251, D0 and
MOVE.B #0x1251, D0

Question 4

Write the sequence of instructions that perform the following operation:
(a) Put 42 into memory location 0x7000;
(b) Put 42 into data register D2;
(c) Move the least significant byte in data register D2 to memory location 0x7000
Question 5

Suppose that the CPU32 registers contains,

(A3) = 0x0001 2343
(D0) = 0x0000 0000
(D5) = 0x3333 5555

and the memory location,

(012343) = 0x9A
(012344) = 0x78
(012345) = 0xC2
(012346) = 0x01

Determine the content of registers A3, D0 and D5 and memory locations after executing the following instructions:

MOVE.L A3, D5
MOVE.W (A3), D0
ADD.B 0x012346,A3
CLR.B -(A3)
MOVE.B 3(A3),D0
ADD.B (0x1,A3,D0.B),D5
Question 1
What sequence of instructions performs the operations (assume 8-bit operand):

\[(0xFFF) + (0xABC) \rightarrow (0xABC)\]

Question 2
Write the sequence of instructions that performs the operations:

\[
\begin{align*}
1 + (0x5000) & \rightarrow (0x5000) \\
2 + (0x5001) & \rightarrow (0x5001) \\
3 + (0x5002) & \rightarrow (0x5002)
\end{align*}
\]

Question 3
Five 16-bit numbers are stored in consecutive memory locations starting at address 0x5000. For each number, the high-byte is stored first then the low byte. Write a program that will add the five numbers and store the sum as a 16-bit number at memory location 0x5100.

Question 4
What sequence of instructions performs the multiplication (assume 8-bit operand):

\[(0x5000) \times (0x8000) \rightarrow (0x7000)\]

Question 5
Write the sequence of instructions that performs the operation:

\[
\text{if } ((A) = 1) \text{ or } ((A) = 2) \text{ then } (A) \rightarrow (B) \text{ else } (B) \rightarrow (A)
\]

Assume that A and B are memory locations.

Question 6
Write a program segment that will subtracts the byte in memory location 0x3124 from the byte in memory location 0x5555 and store the result in memory location 0x3456.

Question 7
(a) What instruction rotates the contents of D0 five step to the right? Assume a 32-bit operand.

(b) What instruction rotates the content of D0 two step to the left? Assume 8-bit operand.

Question 8
What shift instruction can be used to implement multiplication by two of a memory word stored at location 0xNUM (assume 16-bit operand)?

Question 9
What shift instruction can be used to implement integer division by two of a memory word stored at location 0xNUM (assume 16-bit operand)?
Question 1
In C there is a function 'strcpy' that copies one string to another. Strings are terminated with the 0x00 ASCII character. A possible definition in C is:

```c
strcpy(char *dest, char *source)
{
    do{
        *(dest++) = *source;
    }while(*(source++));
}
```

Implement strcpy in assembly language, based on the following shell (not a subroutine).

Note that the 0x00 character must be copied too (the C code does).

```
STRCPY: ; put your code here
        ; put your code here
.section .data
SOURCE: .asciz "CT105"
DEST:    DS.B 11; reserves 11 bytes of space
```

Question 2
Modify your STRCPY code so that it becomes a subroutine. Use A1 and A2 as inputs, with A1 = source address and A2 = destination address.

Question 3
Write a subroutine that converts all characters in a string to uppercase. Assume that the input to the subroutine, the address of the string, is in the A0 register. A crude way of converting to uppercase is to AND the character with 0x5F, although this will cause error in some ASCII characters such as digits. Can you improve your subroutine to check that the character is between 'a' and 'z' before converting to uppercase?

Question 4
Write a subroutine,"MAX" (inputs - D0 and D1) that determines which of the two 32-bit unsigned integers in D0 or D1 are the largest. D0 is assigned the largest of these numbers.

Question 5
Use the subroutine in the previous exercise to write a program that determines the largest of all vector elements contained in \( \text{Vec}[i] = 0, 1, 2, 3, ..., N-1 \), where all \( N \) vector elements are 32-bit unsigned numbers and stores the maximum value in D0.
Question 1
We shall perform calculations using the stack which is maintained by A0 as a stack pointer. Implement the following subroutines:

**ENTER:** Pushes the content D0 onto the stack
**ADDSSTACK:** Pops the top of the stack item and adds it to D0
**SUBSTACK:** Pops the top of stack item and substracts it from D0
**POPPSTACK:** Pops the top of stack item into D0

Question 2
Consider the following program which uses the subroutines in previous exercise

MOVEA.L #0x8000, A0
MOVE.L #0x10000005, D0
BSR ENTER
MOVE.L #0x10000006, D0
BSR ENTER
BSR ADDSTACK
BSR SUBSTACK

Show the content of the stack (and A0) after the execution of each subroutine.

Question 3
In the example program below, we show the addresses to the left. Analyse the program and answer the questions below:

8000 BSR NSUM
8004 MOVE.L #3, D0
8008 NSUM: MOVE.L D0, -(SP)
8010 MOVE.L #0, D1
8016 SUBI.L #1, D0
801A BEQ OUT
801C BSR NSUM
801E ADD.L D0, D1
8020 OUT: MOVE.L (SP)+, D0
8020 RTS

(a) Assume that (SP) = 9000 initially. Show the content of the stack and the stack pointer each time SUBI.L is executed.
(b) What does D1 contain when the subroutine NSUM has been executed if (D0) = 0x3, initially?

Question 4
In the program below, PREGS pushes all data registers denoted by the word immediately following the subroutine call instruction in the following way: if bit $i = 1$ then $D_i$ is pushed. PREGS also ensures that the return address is the address of the instruction immediately following the word (that is, MOVE.L D1, D2). Write a subroutine that implements PREGS.

JSR PREGS
DC.W %0000000000100101 | D5, D2 and D0 are pushed
MOVE.L D1, D2
Question 5

One efficient method of passing parameters to subroutine is through the system stack. It allows space to be allocated for passing parameters only when needed. After the subroutine is executed, the allocated stack space can be released for other uses.

Reimplement the following routine to use the stack for passing parameters.

```
main:
    MOVE.L #arraya, partbl       | setup A table to pass parameters
    MOVE.L #sizea, partbl+4
    MOVE.L #maxa, partbl+8
    LEA partbl, A0
    JSR FIDMAX
    MOVE.L #arrayb, partbl      | setup A table to pass parameters
    MOVE.L #sizeb, partbl+4
    MOVE.L #maxb, partbl+8
    LEA partbl, A0
    JSR FIDMAX
    MOVE.W maxa, D0
    CMP.W maxb, D0
    BGE STORE
    MOVE.W maxb, D0
    STORE: MOVE.W D0, max
    RTS

FIDMAX: MOVEM.L D0/D7/A1/A2, -(SP) | Save D0,D7,A1,A2 onto the stack
    MOVEA.L (A0), A1          | A1 = Address array a
    MOVEA.L 4(A0), A2         | A2 = Address of array size
    MOVE.W (A2), D7           | D7 = Array size
    MOVE.L 8(A0), A2          | A2 = Address to store max value
    SUBI.L #2, D7             | Use D7 as loop counter
    MOVE.W (A1)+, D0          | Use D0 to store maximum value

LOOP:
    CMP.W (A1)+, D0
    BGE ENDLOOP
    MOVE.W -2(A1), D0

ENDLOOP:
    DBF D7, LOOP
    MOVE.W D0, (A2)
    MOVEM.L (SP)+, D0/D7/A1/A2
    RTS

.section .data
sizea:  DC.W  10
maxa:   DS.W   1
arrayb:  DC.W   65, -22, 35,11, 77, 55, -11
sizeb:  DC.W   7
maxb:   DS.W   1
max:    DS.W   1
partbl: DS.L   3
```
Question 1

Consider the following 'C' program,

```c
main()
{
    int x, result;
    x = 8;
    result = double(x);
} /* main() */
```

Implement the `double()` (result = 2 * x) routine in assembly language, based on the following shell.

```
.globl double

double:
    ; put your code here
    ; put your code here
    RTS
```

Question 2

Consider the following 'C' program,

```c
main()
{
    int vector[5], counter;
    int summation;

    vector[0] = 8; vector[1] = 12;
    vector[4] = 27;

    counter = 5;

    summation = sumv(vector,counter);
} /* main() */
```

Implement the `sumv()` routine in assembly language, based on the following shell,

```
.globl sumv

sumv:
    ; put your code here
    ; put your code here
    RTS
```
Question 3

Consider the three-bus architecture of the MC68332 and most microprocessors.

(a) Name the three buses and explain how each bus is used.

(b) Draw a diagram to show how the three buses achieve communication between parts of a microprocessor system.

Question 4

An 8K x 8 memory chip (i.e., 0x2000 addresses, 1 byte each) is to be connected to the MC68332 buses in such a way that it occupies memory addresses 0x6000 to 0xFFFF inclusive (RAM base address is 0x6000). Design the chip select logic for this configuration. Note that the memory chip will need to be selected when a valid memory address is identified, regardless whether it is writing or reading. The chip will have extra signals to distinguish read and write.

(a) Using the traditional address decoding method.

(b) Using the modern way (using chip select 5 via software).

Question 5

What signal will be present on the three microprocessor buses when the CPU is executing the following instruction,

\[
\text{MOVE.W} \ #0x34, \ \text{RAM} + 0xFF
\]